Space Flight Middleware: Remote AMS over DTN for Delay-Tolerant Messaging

Scott C. Burleigh[1]
Jet Propulsion Laboratory, California Institute of Technology, Pasadena, California, 91109

This paper describes a technique for implementing scalable, reliable, multi-source multi-point data distribution in space flight communications – Delay-Tolerant Reliable Multicast (DTRM) – that is fully supported by the "Remote AMS" (RAMS) protocol of the Asynchronous Message Service (AMS) proposed for standardization within the Consultative Committee for Space Data Systems (CCSDS). The DTRM architecture enables applications to easily "publish" messages that will be reliably and efficiently delivered to an arbitrary number of "subscribing" applications residing anywhere in the space network, whether in the same subnet or in a subnet on a remote planet or vehicle separated by many light minutes of interplanetary space. The architecture comprises multiple levels of protocol, each included for a specific purpose and allocated specific responsibilities: "application AMS" traffic performs end-system data introduction and delivery subject to access control; underlying "remote AMS" directs this application traffic to populations of recipients at remote locations in a multicast distribution tree, enabling the architecture to scale up to large networks; further underlying Delay-Tolerant Networking (DTN) Bundle Protocol (BP) advances RAMS protocol data units through the distribution tree using delay-tolerant store-and-forward methods; and further underlying reliable "convergence-layer" protocols ensure successful data transfer over each segment of the end-to-end route. The result is scalable, reliable, delay-tolerant multi-source multicast that is largely self-configuring.

I. Introduction

WHILE many types of space flight system communications are intended for reception by a single specified communicating entity, such as the command and data handling system of an orbiting spacecraft, others may be most effective if directed to all members of a possibly dynamic group of interested entities. For example, it might be advisable for spacecraft health alerts to be delivered to an on-board flight recorder task for local logging and also delivered to both an analytical engine and an operator's console display on Earth.

Such "multipoint" communications are commonly implemented in terrestrial networks by the use of multicast protocols (notably IP multicast) and/or by messaging "middleware" systems that implement a "publish/subscribe" transmission model (such as the Java Message Service). But for the operations of spacecraft, especially flight vehicles in deep space, neither of these approaches is optimal: scalable, reliable multi-source multicast remains a research problem, and existing messaging middleware systems rely on underlying network protocols that don't work well over the punctuated connectivity and long signal propagation latencies of flight operations.

This paper describes an alternative approach – Delay-Tolerant Reliable Multicast (DTRM) – that is fully supported by the "Remote AMS" (RAMS) protocol of the Asynchronous Message Service (AMS) proposed for standardization within the Consultative Committee for Space Data Systems (CCSDS). The DTRM architecture enables applications to easily "publish" messages that will be reliably and efficiently delivered to an arbitrary number of "subscribing" applications residing anywhere in the space network, whether in the same subnet or in a subnet on a remote planet or vehicle separated by many light minutes of interplanetary space.

A. Motivation

The "Mission Operations Services Concept" report published by CCSDS[1] summarizes the advantages of basing flight mission operations on a standardized "service oriented" communications architecture. Among these are enhanced interoperability among space agencies (potentially reducing mission risk); reduced cost and risk due to re-usability of proven infrastructure; further cost savings from more direct competition among equipment suppliers;

[1] Principal Engineer, Systems Engineering, 4800 Oak Grove Drive, m/s 301-490.

and enhanced performance due to easier reallocation of mission functions between flight and ground systems. The report goes on to identify messaging "middleware" as an enabling technology for such an architecture.

These findings echo the growing acceptance of service-oriented architectures in financial, commercial, and industrial network communications, e.g. Ref.2, and the value of multicast techniques in support of data delivery to large numbers of users – both in commerce and in military communications (e.g., Ref. 3) – is widely acknowledged.

B. Current Approaches

Standards for reliable multi-point delivery that scales up to very large numbers of terrestrial users are not yet universal, however, and the problem is especially challenging in a space flight environment. Flight communications frequently are characterized by routine interruption of end-to-end connectivity (as when satellites pass out of view of ground stations), high noise levels resulting in non-congestion-related data loss, and – in the extreme – lengthy signal propagation delays as data traverse interplanetary distances on the order of light minutes or hours. These characteristics of space flight network communications make the use of middleware and multicast technologies developed for the Internet (e.g., Java Message Service[4], WebSphere Message Broker[5], and Advanced Message Queuing Protocol[6]) impractical: such technologies rely on Internet infrastructure that includes automated routing protocols built on TCP, but this infrastructure is unsuitable for space flight missions because TCP performs poorly over space links[7].

The challenges of reliable unicast over a network that functions efficiently in the space flight environment are addressed by the Delay-Tolerant Networking (DTN) architecture[8-12], introduced by the DTN Research Group of the Internet Research Task Force. A wide variety of multicast architectures for DTN have also been proposed (e.g., Refs. 13-15) but DTN multicast must overcome severe obstacles in order to achieve scalable reliability. In particular, the retransmission system built into the DTN Bundle Protocol[9] (BP), "custody transfer", is not designed to support a branching tree of bundle custodians.

II. Architecture

A. DTRM Network Stack

To solve this complex problem we decompose it into smaller, individually simpler problems that are susceptible to a modular solution. That is, DTRM is based not on a single new protocol that addresses all of the challenges of space flight middleware but rather on the integration of multiple existing protocols that address those challenges individually. Well-defined interfaces facilitate the "stacking" of these individually efficient and proven protocols into an aggregate capability.

But beyond the straightforward and familiar stacking of protocols, DTRM is actually built on a *stack of networks*. Each network in the stack is an overlay superimposed on the network(s) below it, playing a well-defined role in the end-to-end architecture.

1. Subscriptions

The uppermost networks in the DTRM stack are DTRM *subscriptions*. Subscriptions are simple "star" networks that – in concept – directly convey the messages exchanged between message publishers and subscribers (Fig. 1). Each DTRM subscription comprises a single application *module* (i.e., a process, task, or thread) – the "hub" of the star – that subscribes to messages on a given topic, together with all other modules that may publish those messages.

Each DTRM subscription is instantiated as a CCSDS Asynchronous Message Service[16] (AMS) subscription formed by the configuration message traffic of the CCSDS Meta-AMS (MAMS) and Remote AMS (RAMS) protocols (idem.):

- The subscribing module issues MAMS messages (subscription assertions and cancellations) directly to other modules in the same AMS *continuum* (that is, local network), typically using delay-sensitive protocols.
- When potential sources of messages on the subscription topic reside in a remote continuum (e.g., aboard a spacecraft in interplanetary space), a RAMS *gateway* module located in the local continuum uses delay-tolerant transmission protocols to forward the MAMS information to its peer gateway module in the remote continuum, which in turn forwards it to the other modules in that continuum.

Note that the union of all DTRM subscriptions whose subscribing modules subscribe to messages on any single topic – termed a DTRM *group* – is functionally analogous to the membership of a single IP multicast group (with the message topic serving the same function as an IP multicast address) except that the DTRM group is not limited to a single message source. Figure 2 depicts the DTRM group formed by the subscriptions in Figure 1.

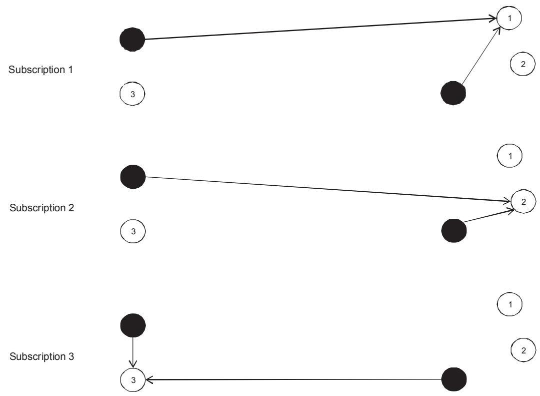

Figure 1. Three DTRM subscriptions on a common message topic. *Each subscription is a simple "star" network with the subscriber at its hub.*

The union of all subscriptions whose subscribing modules subscribe to *any* topic in the performance of some common instance of a single distributed application – analogous to a set of related multicast groups – is termed a DTRM *venture*.

2. Distribution Tree

Each DTRM subscription's operations are implemented by the flow of messages through an underlying DTRM *distribution tree*. That is, while DTRM subscription topology is star-shaped in the abstract, it is in practice mapped to the nodes of a tree, making it more scalable than it might otherwise seem; see Fig. 3 and Fig. 4.

A DTRM distribution tree is functionally analogous to the distribution tree of an IP multicast group, except that while each IP multicast group must have its own distribution tree, a single DTRM distribution tree can support *all* of the DTRM subscriptions (that is, all groups) in any single DTRM venture. This is because the venture's DTRM distribution tree is implemented as the *RAMS network* for that venture, i.e., the set of all RAMS gateways in all the networks participating in the venture, and a single RAMS gateway can forward the application and group configuration messages for any number of application message topics.

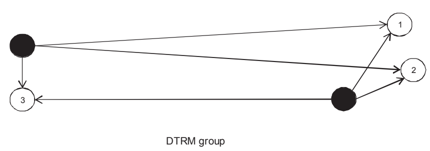

Figure 2. The DTRM group formed by these subscriptions.

3. DTN Mesh

Each DTRM distribution tree is an overlay on a partial mesh network implemented by the Delay-Tolerant Networking Bundle Protocol (Fig. 5).

4. Internets

Each DTN mesh is itself an overlay over the concatenation of one or more possibly heterogeneous underlying networks, here termed *internets*, at what the DTN architecture identifies as the "convergence layer". Each network at the convergence layer of the DTRM stack may be an IP-based network – either the worldwide Internet or a private local area network such as might reside on a spacecraft – but might equally be a network built on other protocols or a wholly private communications infrastructure such as a single radio link; see Fig. 6.

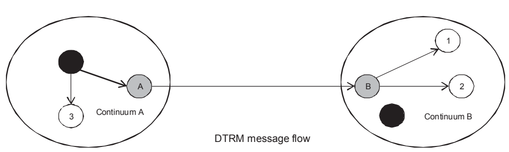

Figure 3. Message flow through the DTRM group, for a single message publication. *Although there are three subscribers, the publishing node issues only two copies of the message.*

5. Subnets

And of course an IP-based internet is in turn an overlay over the concatenation of one or more possibly heterogeneous subnets.

The advantage of stacking these networks one upon another is analogous to the advantage of protocol layering in any single network: different functions can be allocated to the different layers, enabling each to be implemented independently and relatively simply while preserving efficient operation of the stack as a whole:

- The AMS-based subscriptions implement multi-point data delivery.
- The underlying distribution tree protects scalability.
- The DTN mesh provides store-and forward routing that is tolerant of delay and disruption.
- The local protocols implementing the convergence-layer internets ensure reliable data transmission individually on each segment of the end-to-end DTN path from each publisher to each subscriber.

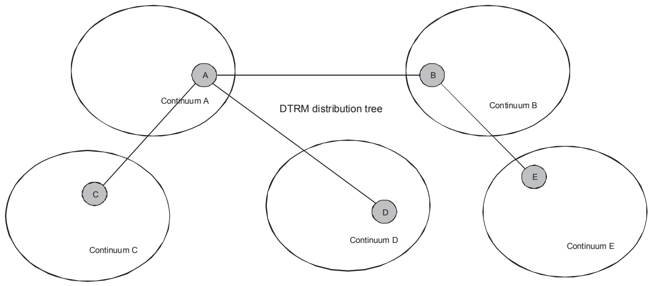

Figure 4. The underlying DTRM distribution tree.

B. How DTRM Differs From Other Multicast Architectures

IP multicast is based on just three layers of network: the underlying Internet, its underlying subnets, and a single overlay, the distribution tree of an IP multicast group. The nodes of the distribution tree are a subset of the nodes of the Internet, and the distribution tree nodes are themselves of two classes: group members (the leaves of the tree, including the source of the multicast packets) and forwarding nodes (the non-leaf nodes of the tree). Multicast routing protocols are used to form the distribution tree of a multicast group dynamically and automatically when it is created and to revise that tree as group membership changes. Note that any change in either the membership of the group or the topology of the underlying internet must modify the multicast group's distribution tree.

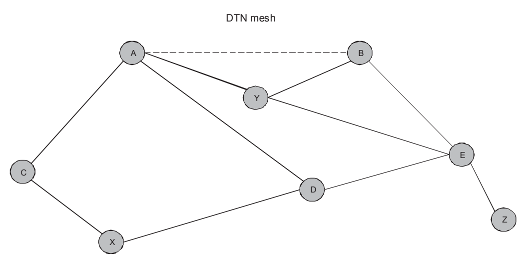

Figure 5. The DTN mesh on which the distribution tree is overlaid. *Note that the notional connection from gateway A to gateway B is in fact an indirect route through DTN node Y, which is not a DTRM gateway.*

The DTN multicast architectures proposed to date take essentially the same approach, with an extra layer of network (the DTN mesh) inserted. The nodes of the DTN mesh are a subset of the nodes of the underlying internets; the nodes of the DTN multicast distribution tree are a subset of the nodes of the DTN mesh; and the distribution tree nodes are again of two classes: group members (the leaves of the tree, including the source(s) of the multicast bundles) and forwarding nodes (the non-leaf nodes of the tree). Again, a multicast routing protocol [yet to be developed] is needed in order to form the distribution tree of a new DTN multicast group. And, again, any change in either the membership of the group or the topology of the underlying DTN must modify the multicast group's distribution tree.

In contrast, DTRM allocates the functionality of the classical multicast distribution tree to two layers of network rather than one:

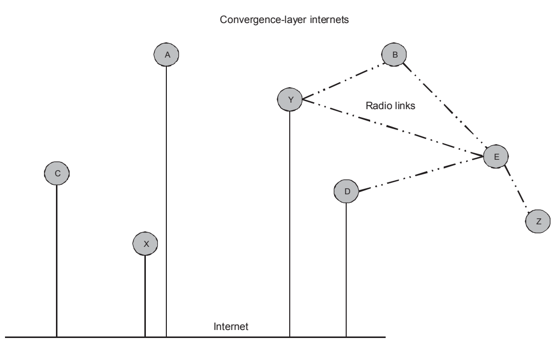

Figure 6. The convergence-layer internets on which the DTN mesh is overlaid. *The DTRM design ensures that only a single copy of the published message shown in Figure 3 traverses the radio link to the subscribers in continuum B.*

- A DTRM group includes all AMS modules that subscribe to – or publish – messages on the topic of this group.
- A DTRM distribution tree comprises all RAMS gateways in the DTRM venture. As such it provides the forwarding structure for operating an unlimited number of groups, because RAMS gateways are not topic-specific: each one can forward the application and group configuration messages for any number of topics.

A change in the membership of a DTRM group therefore affects the associated subscription star(s), but it has no effect on the underlying DTRM distribution tree's topology. Only a change in the topology of the underlying DTN mesh will change the DTRM distribution tree.

Moreover, the design of RAMS ensures that no change in the membership of a DTRM group is propagated any further than necessary: each RAMS gateway acts as the "agent" for all group members in its local continuum, so (for example) each subscription assertion after the first is merely noted in the state of the gateway rather than forwarded to other gateways.

This design enables enhanced flexibility in application configuration while reducing the network bandwidth consumed by protocol overhead:

- Group members can be added or deleted rapidly without generating any multicast routing protocol traffic, because the topology of the distribution tree is unaffected.
- Even entire new groups can be added or deleted rapidly, because there is never any need to build an additional distribution tree: the distribution tree that is common to all existing groups automatically supports an unlimited number of new ones as well.

The ease with which new subscription stars can be generated, without incurring multicast routing protocol overhead, makes multi-source multicast in DTRM straightforward. At the same time, the structure of the underlying DTRM distribution tree offers the same network bandwidth efficiency that characterizes conventional multicast:

- When, for example, a module publishes a message for which there are two local subscribers and 80 subscribers in other continua, only three copies of the message are issued by the publishing node itself: one to each local subscriber and one to the RAMS gateway module.
- When the gateway module receives that message and (in light of prior configuration traffic) recalls that there are subscribers to the topic of that message in continua served by two of its neighboring gateways in the RAMS network – but not to its other three gateway neighbors – it forwards one copy of the message to each of those two gateways.
- When one of those remote gateways receives its copy of the message, it forwards a copy to each of the subscribers to that message topic in its local continuum and additionally forwards copies to the other gateways who have announced their own interest in this message topic, and so on.
- Eventually, copies of the message are delivered to all 80 subscribers, but the number of copies that traverse the underlying DTN mesh is far fewer than 80.

III. Implementation

A. Implementations of DTRM Components

Open-source implementations of all of the protocols utilized in the DTRM architecture are freely available. The "ION" implementation of the DTN stack, including implementations of the AMS protocols, can be downloaded from http://www.openchannelfoundation.org/projects/ION.

The AMS protocols have also been implemented by developers at NASA's Marshall and Goddard Space Flight Centers and at the SciSys Group, United Kingdom.

Other implementations of the DTN protocols have likewise been developed by a variety of research organizations. Most prominent among these is the DTN reference implementation "DTN2"; sources for DTN2 and other implementations are provided at http://www.dtnrg.org/wiki/Code.

B. Operational Experience

Standardization of the protocols utilized in the DTRM stack is not yet final, but preliminary testing of prototypes has been encouraging.

An early version of the DTRM stack was exercised in the spring of 2006 over a simple DTN mesh encompassing nodes at the Jet Propulsion Laboratory (JPL) in Pasadena, CA; at California Polytechnic State University, San Luis Obispo; at the Johns Hopkins Applied Physics Laboratory (APL) in Columbia, MD; and at the Marshall Space Flight Center in Huntsville, AL. The testing was hampered by a number of errors discovered in the software and in the configuration of the communicating nodes, but eventually sustained DTRM traffic – clocked at about 7.4 Mbps – was established between JPL and APL[17].

DTRM traffic first flowed over interplanetary links in October of 2008, during the first Deep Impact Network (DINET) experiment. More mature implementations of the DTRM protocols were used to convey published messages from two JPL computers to a third – to which the others had no direct connectivity – via a Bundle Protocol router node residing on the EPOXI spacecraft, at a distance of from 49 to 81 light seconds from Earth. EPOXI was at that time in an inactive cruise period while en route to encounter comet Hartley 2 (November 2010). Communication with the spacecraft was limited to eight Deep Space Network tracking passes of four hours each, separated by intervals of two to five days. The uplink data rate to the spacecraft was limited to 250 bytes/second, while the downlink data rate from the spacecraft was normally about 20000 bytes/second. In all, 292 images (about 14.5 MB) were conveyed through the DTRM group, with no data loss or corruption detected anywhere in the network[18].

IV. Conclusion

Final approval of the AMS protocols as CCSDS Recommended Standards is anticipated sometime in the first half of calendar 2011. At that point, specifications for all components of the DTRM stack will be available as open, published documents backed by open-source implementations. DTRM capability will be offered to the designers of flight missions for all national space agencies, but DTRM may additionally have utility in terrestrial applications that could benefit from scalable, reliable, disruption-tolerant, multi-source multicast.

Acknowledgments

The research described in this paper was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

References

[1]Consultative Committee for Space Data Systems, "Mission Operations Services Concept", CCSDS 520.0-G-3, December 2010.

[2]Computer Business Review, "Oracle launches Application Integration Architecture Release 3.1", *CBR Online*, 17 February 2011, URL http://middleware.cbronline.com/news/oracle-launches-application-integration-architecture-release-31-160211 [cited 28 February 2011].

[3]Choudhary, M., Sharma, P., and Sanghi, D., "Secure Multicast Model for Ad-hoc Military Networks", *Proceedings of 12th IEEE International Conference on Networks*, vol. 2, Singapore, 16-19 November 2004, pp. 683 - 688.

[4]Chappell, David A., and Monson-Haefel, Richard, *Java Message Service*, O'Reilly Media, Sebastopol, CA, December 2000.

[5]Piper, Andy, "WebSphere Message Broker: Designing for Performance", *IBM Software Conference Impact 2010*, Las Vegas, 2-7 May 2010.

[6]Lai, Eric, "Open standard messaging protocol may challenge IBM's WebSphere MQ", *Computerworld*, 21 June 2006.

[7]Akan, O.B., Fang, H., and Akyildiz, I.F, "Performance of TCP Protocols in Deep Space Communication Networks", *IEEE Communications Letters*, Vol. 6, No. 11, November 2002, pp. 478-480.

[8]Cerf, V., Burleigh, S., Hooke, A., Torgerson, L., Durst, R., Scott, K, Fall, K., and Weiss, H., "Delay-Tolerant Networking Architecture", Internet Research Task Force, RFC 4838, April 2007.

[9]Scott, K, and Burleigh, S., "Bundle Protocol Specification", Internet Research Task Force, RFC 5050, November 2007.

[10]Burleigh, S., Ramadas, M., and Farrell, S., "Licklider Transmission Protocol – Motivation", Internet Research Task Force, RFC 5325, September 2008.

[11]Ramadas, M., Burleigh, S., and Farrell, S., "Licklider Transmission Protocol – Specification", Internet Research Task Force, RFC 5326, September 2008.

[12]Farrell, S., Ramadas, M., and Burleigh, S., "Licklider Transmission Protocol – Security Extensions", Internet Research Task Force, RFC 5327, September 2008.

[13]Zhao, W., Ammar, M., and Zegura, E., "Multicasting in Delay Tolerant Networks: Semantic Models and Routing Algorithms", *Proceedings of the 2005 ACM SIGCOMM Workshop on Delay-Tolerant Networking*, Philadelphia, 21-26 August 2005.

[14]Abdulla, M, and Simon, R., "A Simulation Analysis of Multicasting in Delay Tolerant Networks", *Proceedings of the 2006 Winter Simulation Conference*, Monterey, California, 3-6 December 2006.

[15]Ye, Q., Cheng, L., Chuah, M. C., and Davison, B. D., "Performance Comparison of Different Multicast Routing Strategies in Disruption Tolerant Networks", *Computer Communications*, Vol. 32, No. 16, 15 October 2009, pp. 1731-1741.

[16]Consultative Committee for Space Data Systems, "Asynchronous Message Service", CCSDS 735.1-B-0, March 2010 (pending publication).

[17]Burleigh, S., "Report on Wide-Area Network Testing of the CCSDS Asynchronous Message Service 31 March 2006 through 26 May 2006", JPL internal report, 30 May 2006 (unpublished).

[18]Jay Wyatt, E. J., Burleigh, S., Jones, R., Torgerson, J. L., and Wissler, S., "Disruption Tolerant Networking Flight Validation Experiment on NASA's EPOXI Mission", *Proceedings of the 2009 First International Conference on Advances in Satellite and Space Communications*, Colmar, France, 20-25 July 2009.